LA-UR 99-1233

Quality Assessment and Restoration of Typewritten Document Images

Michael Cannon, Judith Hochberg, and Patrick Kelly
Los Alamos National Laboratory

Mailing Address:
Michael Cannon
Mail Stop B265
Los Alamos National Laboratory
Los Alamos, NM 87544
tmc@lanl.gov

Quality Assessment and Restoration of Typewritten Document Images

Michael Cannon, Judith Hochberg, and Patrick Kelly
Los Alamos National Laboratory

Abstract. We present a useful method for assessing the quality of a typewritten document image and automatically selecting an optimal restoration method based on that assessment. We use five quality measures that assess the severity of background speckle, touching characters, and broken characters. A linear classifier uses these measures to select a restoration method. On a 139-document corpus, our methodology reduced the corpus OCR character error rate from 20.27% to 12.60%.

Key words: Optical character recognition - Document quality assessment - Document image restoration.

1. Introduction

Not all of today's OCR is performed on clean laser-written documents. Many organizations have huge archives of typewritten material, much of it of marginal quality. For example, the U.S. Department of Energy has an archive of over 300 million classified documents consisting of fixed-width, fixed-pitch typewritten documents, teletypewriter output, and carbon copies on aging fibrous paper. As part of the declassification review process, almost all of these documents have been photocopied and/or photoreduced. By today's OCR standards, this archive and others like it are of marginal quality. Even though successful document enhancement methods are known [1 - 3], they must often be applied under human guidance to avoid further image degradation. However, when dealing with a document archive of this size, it is not humanly practical to select the best restoration method for each document.

In this paper we therefore present a method for *automatically* improving the quality of document images in such an archive, and we demonstrate a marked decrease in OCR error rates. The method consists of two parts. First, we use five measures to assess the quality of a document image. Second, we

use this quality assessment to automatically select an optimal restoration algorithm for each document by means of a linear classifier.

Our approach is reminiscent of earlier work [4, 5] in its assessment of document quality prior to restoration. It differs in its focus on typewritten text instead of handwriting, and its automatic choice of restoration. Our quality measures assess the extent of background speckle, broken characters, and touching characters. We did not concern ourselves with document cutoff or skew [4].

The remainder of this paper is organized as follows. Section 2 describes our data corpora. Sections 3 and 4 describe the two analytic threads underlying the classifier: the quality measures and the restoration algorithms. Section 5 describes the classifier and its efficacy, Section 6 speculates about a cascade of restoration methods, and Section 7 presents some conclusions.

2. Data

The Department of Energy made a 139-member 300-dpi corpus of document images available to us for this work. Its quality is representative of the archive mentioned above. The text is horizontal with near-zero skew and is in simple single-column layouts. Ground-truth text files for the documents were also made available. We used Caere OmniPage Pro v8.0 to perform OCR and found the character error rate of the corpus to be 20.27%. We were curious how our approach would perform on an especially degraded corpus of images. We therefore formed a subcorpus of 41 documents having OCR character error rates between 20% and 50%, averaging 31.28%, with a word error rate of 50.99%. We did not include documents with character error rates higher than 50% because we did not trust the ability of string-matching algorithms in that range to give us the accurate error rate required by the linear classifier.

In order to create document quality measures under controlled conditions, we created a second, very small corpus of documents spanning a range of gradually decreasing document image quality. We did this by repeatedly photocopying a page from a book, so that we had nine versions of it, the original and eight following photocopy generations. Each successive generation was increasingly plagued with common attributes of lower quality document images: background speckle, fattened stroke-widths, and touching characters.

3. Quality Measures

Our document image quality measures are designed to quantify the document degradations we observed in the DOE corpus and our photocopied corpus. Many of these degradations are illustrated in Figure 1, which is a portion of the eighth-generation document from the smaller corpus, and in Figure 2, which is from the larger DOE corpus.

We formulated five quality measures, each normalized to the range 0 to 1. The following is a preliminary description of the measures. A more technical definition will follow in section 3.2.

1. *Small Speckle Factor (SSF).* The small speckle factor measures the amount of black background speckle in the document image. The origin of the speckle varies. The speckle in Figure 1 is strictly an artifact of our copying machine. In our larger DOE corpus, much of it arises from photocopying low contrast documents (Figure 2). The background speckle can sometimes be so severe that it is interpreted as text by the OCR engine.
2. *White Speckle Factor (WSF).* Many degraded documents exhibit fattened character strokes. This problem can arise in carbon copies of documents, especially photocopies of carbon copies. The fattened stroke width can lead to OCR difficulties by creating unexpected small white connected components or by reducing or eliminating expected white components. Both of these problems can be seen in the word “lesser” in Figure 1, where the white component in the upper half of the second “e” shrinks and a new component is created in the lower half of the “e”. An unexpected white connected component can also be seen in both occurrences of the letter “s” in Figure 2.
3. *Touching Character Factor (TCF).* The touching character factor measures the degree to which neighboring characters touch. Like white speckle, touching characters are caused by fattened strokes, as seen in the word “ventricular” in Figure 1. Touching characters cause problems for OCR by making it difficult to differentiate between certain letters such as “ni” and “m”, and by creating completely novel and uninterpretable text.

4. *Broken Character Factor (BCF).* The broken character factor measures the degree to which individual characters are broken. Broken characters often arise from photocopying low contrast documents, as seen in both occurrences of the letter "e" in Figure 2.
5. *Font Size Factor (FSF).* We find a correlation in our corpus between OCR accuracy and the size of the font. This correlation might not stem from the font size *per se*, but rather from degradations that accompany an increase or decrease in the size of the font.

How meaningful are these measures? To address this question, we computed the correlation between our quality measures and the OCR character error rate of our 139-document corpus. The results are presented in Table 1. The right-most column shows that the quality measures are indeed correlated with the OCR error rate. In fact, some of our earlier ideas for quality measures, such as a *Large Speckle Factor,* were discarded if they correlated poorly with the error.

As we developed the five quality measures, some of the parameters within each measure were tweaked in order to make the correlation with the OCR error as high as possible. The high .681 correlation between the white speckle factor and the error results from that effort. The low .173 correlation between the broken character factor and the error obtains *in spite* of that effort! We also tweaked parameters in order to make the correlation *between* measures as small as possible. The low .176 cross-correlation factor between the small speckle factor and the broken character factor represents that effort. The high .649 cross-correlation between the touching character factor and the white speckle factor obtains *in spite* of that effort. In this case, the high correlation is understandable, as both factors indirectly measure the fattening of stroke width.

In the end, we found a strong linear relationship between our quality measures and the OCR error rate. Using the standard linear least squares method to model the relationship, we designed a linear predictor using half of the data, and then tested it using the other half. The correlation between the actual OCR error rates and the predicted ones was .894, an indication that the quality measures are indeed meaningful.

3.1 Analyses Used in Computing Quality Measures

This subsection lays the groundwork for a more technical definition of the five quality measures. We will describe the analyses of connected components, text lines, and font size that will be used to define the measures in section 3.2.

The small speckle factor and white speckle factor make use of histograms of black and white connected components in a document [6]. For each document, we find these connected components based on eight-connectivity, and compute separate histograms for the black and the white components. This process is illustrated in Figures 3 and 4, which show the histograms for the zeroth and fifth generation photocopies in our smaller corpus. Between these generations, the number of small black connected components increased due to the greater incidence of background speckle (Figure 3). At the same time, the number of small white connected components increased as fattened black strokes created new white components and caused existing ones to shrink (Figure 4).

While they are not exploited in our current set of quality measures, further features of these histograms show the main character lobe shifting rightward, and also show the increasing incidence of touching characters, both due to fattening stroke widths.

All the quality measures are dependent on the size of the font used in the document. Even though most of the documents in our DOE corpus are typewritten, some are large-font teletype-written documents, and some documents have been photoreduced. Our method for finding font size[1] is based on an analysis of its black connected components. We first compute the histogram of the heights of the bounding boxes of these components. A typical histogram is shown in Figure 5. The histogram is usually bimodal, if we ignore the first few bins that are sometimes influenced by black background speckle.

The peak at 25 pixels represents the x-height of a normal character, and is the one we want. The next peak represents characters with an ascender or descender. We define a peak as a bin having a value greater than its 4 neighbors. The first peak we encounter, working from the left, that is greater than 10% of the total area under the histogram is declared to be the x-height of the font in pixels. In order to skip over a possible peak resulting from black background speckle, we ignore the first ten bins of the

histogram. A hand-check revealed that the algorithm made no mistakes in our 139-member corpus. We refer to the pixel-height of the font as the "font size" in this paper. Typical font size values in our corpus range from 10 to 40 pixels. We notice that the main character lobe of black connected component sizes for our data set (Figure 4) does not extend beyond approximately the location "font size squared." We use this fact in Section 3.2 to normalize the quality measures.

Finally, the Touching Character Factor and Broken Character Factor depend on finding text lines. For example, if the computation of the broken character factor were not constrained to lines of text, it would be greatly affected by any background speckle in the document. We find lines of text in a document from the horizontal projection, which we smooth with a median filter of length 3 applied iteratively to convergence. Peaks in the smoothed projection indicate lines of text in the document.

We discovered that false text lines are found in documents that have an inordinate amount of background speckle. In these documents, intraline speckle produces peaks larger than some short lines of text. We remedy this problem by retaining only the tallest 50% of the peaks. Of course, in many cases we throw away many good lines of text, but since we are assessing quality rather than performing OCR, this is not an issue. For a connected component to lie on a "best" text line, both its top and bottom extrema must be contained in the line's horizontal projection.

3.2 Definition of the Quality Measures

Using our knowledge of a document's connected components, the font size, and best text lines, we are in a position to formalize the five quality measures. The following description contains sufficient detail to allow implementation of the measures.

3.2.1 Small Speckle Factor

The small speckle factor measures the amount of black background speckle in the document image. Its computation depends on the document's black connected component histogram and its font size.

---

[1] Because of the photoreduction issues in the corpus, we have avoided use of the more common term "type size".

The amount of speckle is represented by the area under the first lobe in the histogram of black connected component sizes shown in Figure 3. We begin the integration of the area at bin 6 (i.e., black connected components six pixels in size), thus discounting very small speckle that has no effect on OCR accuracy. We end the integration at the bin representing the font size, which is somewhat smaller than the number of pixels in the letter "i". In Figure 6, we illustrate the largest possible pieces of small speckle, ones containing "font size" pixels. Any black connected component larger than this will be considered a character or part of a character. We normalize the measure to the range 0 to 1 by dividing it by the area under the histogram between bin 6 and the bin representing font size squared, which is somewhat greater than the number of black pixels in the letters "o" or "p". We avoid using connected components in the normalization that are much larger than font size squared because that begins to include touching characters, thus deleteriously suppressing the significance of the small speckle factor.

3.2.2 White Speckle Factor

The white speckle factor measures the degree to which fattened character strokes have shrunken existing white connected components and created new ones. Its computation depends on the document's white connected component histogram and its font size.

Our approach is based on the measurement of small white connected components, and is motivated by the white speckle factor presented by Blando, et. al [7]. As the stroke width fattens, the size of small white connected components - such as the one within the letter "e" - shrinks. The increase in the number of these small white components can be seen in the first few bins of the histogram of white connected component sizes shown in Figure 4. We sum the bins between 1 pixel and 1% of font size squared pixels in size. For example, if the font size is 25 pixels in height, the very small white components are those between 1 and 6 pixels in size, as shown in Figure 6. We then normalize the sum by dividing by the total area under the histogram between 1 and font size squared. To avoid measuring white areas in the clumpiness of background speckle, we compute the histogram of white connected component sizes only along the best text lines.

3.2.3 Touching Character Factor

The touching character factor measures the degree to which neighboring characters touch. Its computation depends on the document's black connected components (though not their histogram), its text lines, and its font size.

The key to this measure is realizing that a character that does not touch its neighbors is roughly square. On the other hand, characters that touch are represented by black connected components that are long and low. For this measure, we examine only those black components that lie on the best text lines. We count the number of long and low black components and normalize it by the total number of black components. For a component to be considered long and low, its height-to-width ratio must be less than .75. To avoid speckle and some broken characters, we disregard black components with fewer pixels than 3*(font size) (slightly larger than an "i" or an "l") or shorter than .75*(font size). To avoid big globs of background speckle, we also disregard components taller than 2*(font size). Figure 6 shows the effectiveness of this measure in an example having both broken characters and touching characters.

3.2.4 Broken Character Factor

The broken character factor measures the degree to which individual characters are broken. Its computation depends on the document's black connected components (though not their histogram), its text lines, and its font size.

We make our measurements for this factor only along the best text lines. We count the number of small black connected components (these are broken character fragments) along lines of text and normalize by the total number of components along lines. To be counted as a fragment, a black connected component must be either vertically shorter than or horizontally narrower than 75% of the font size. To avoid measuring small black speckle that may lie on lines of text, we require that a component have more than font size pixels in it. (This is somewhat fewer pixels than the letter "i" contains.). For example, if the font size is 25 pixels, the smallest fragment must have 25 pixels in it and be shorter or narrower than 19 pixels. Figure 6 illustrates the types of black connected component that would figure into the broken character factor.

3.2.5 Font Size Factor

The font size factor is computed by normalizing the font size to the range 0 to 1 and subtracting it from 1. The normalization is based on an assumed font size range in our present corpus between a minimum of 10 and a maximum of 40 pixels. The actual calculation is $FSF = \frac{FS - \min}{\max - \min}$, where *FS* is the font size.

3.3 General Nature of the Quality Measures

We have used several parameters in the definition of the five quality measures. Since each of these parameters is a function of the font size of the document in question, the resulting quality measures are robust and general. As shown in Figure 6, the parameters make intuitive sense. Furthermore, the correlation with OCR error rates shown in Table 1 enhances our confidence in the quality measures. Because they have been developed on a diverse corpus that includes a wide range of font size and quality, the quality measures as detailed above can be used on other document image corpora. (A possible weakness in our methodology, however, is that we have developed the quality measures on the very corpus on which they were tested.) In Section 5.3, we show that the quality measures reflect the improvement obtained by restoring the images in our document corpus.

4. Restoration Methods

Our document image quality restoration methods are designed to repair the degradations addressed by the quality measures. In this section we describe the methods implemented, and the process we used to choose the most useful subset among them.

4.1 The Methods

We implemented eight restoration algorithms. Three algorithms, Global Morphological Close, Despeckle, and kFill filter, were implemented with variations, leading to a total of fourteen methods.

- *Do Nothing*. It may be that the best enhancement for a document image is to leave it alone. Doing nothing is therefore included in our suite of restoration algorithms.
- *Cut on Typewriter Grid*. The documents in our corpus lie on a typewriter (or teletypewriter) grid. If a document is plagued with touching characters, we should in principle be able to separate them if the typewriter grid is known. Our method for determining the typewriter grid is an extension of a method put forth by Lu [8]. We find the typewriter grid by first computing the Fourier transform of the vertical projection of the best text lines. The average of the magnitude-squared of the transforms is computed. A typical average is shown in Figure 7. The prominent peak indicates the period of the typewriter grid. Our cutting algorithm moves along each line of text extracting two neighboring characters at a time. The location of the characters is known from the typewriter grid. We check to see if the two characters constitute the same black connected component - if they do, a white vertical line is drawn between them. If the characters do not touch, the line is not drawn, as it may destroy character detail such as serifs. In order to stay synchronized with the true character positions, the algorithm frequently computes the cross correlation between the typewriter grid and the vertical projection of the line of text and adjusts its position to the point of maximum correlation.
- *Fill Holes and Breaks; Cut*. In order to fill in breaks and fractures in characters, we employ a method described by Loce and Dougherty [9]. The filling operation consists of eroding the character image with 8 simple morphological kernels and ORing the results together. In order to avoid influence from neighboring characters, we operate on each character one at a time by extracting (cutting) it from the typewriter grid.
- *Despeckle; Cut*. In order to suppress black background speckle while preserving character shape, we rely on another method described by Loce and Dougherty [10]. They prescribe a union of a 2-erosion basis set. Each kernel is 3x3 with two nubbins on it, which we apply to one character at a time. We also find that erosion with a single 3x3 morphological kernel can be very effective, and we apply it one character at a time. We call erosion with the single kernel *Despeckle; Cut*. Erosion with the 2-erosion basis set we call *Agressive Despeckle; Cut*.

- *Global Morphological Close*. The noise-suppression attributes of different morphological operations, particularly the close operation, are well known and widely used [11]. For noise suppression, we have included a 3x3 close in our suite of restoration algorithms. This is a traditional implementation applied to the document image as a whole, not character by character. In an attempt to repair breaks in characters, we also include a morphological close with 5x5 and 7x7 kernels.
- *Global Fill Holes and Breaks*. This method for repairing broken and fragmented characters is the same as that described above, except it is applied globally to the document image, not character by character. It is based on the 8-basis set described by Loce and Dougherty [9].
- *Global Despeckle*. This method for suppressing background speckle in a document image is the same as that described above, except it is applied globally to the document image, not character by character. We implemented both versions, *Global Despeckle* and *Global Aggresive Despeckle.*
- *kFill Filter*. "The kFill filter is designed specifically for text images to reduce salt-and-pepper noise while maintaining readability." [12] The kFill filter holds the potential for suppressing background speckle as well as filling small holes and breaks in characters, although it is computationally far more expensive than any of the other restoration methods. We apply it globally to the document image in its 3x3, 4x4, and 5x5 implementations.

4.2 Paring the Methods

With a 139-member corpus, fourteen categories is too many for a statistical classifier to learn. We therefore needed to pare down the list by choosing the most effective methods and eliminating redundancies.

In order to identify the most effective methods, we applied all fourteen methods to each of the 139 document images, OCR'd each of the resulting images, and computed its OCR character error rate. We then counted the documents for which each method resulted in the greatest OCR improvement. This count is shown in the second column of Table 2. Note that some methods, such as Do Nothing, were rarely best for this particular corpus.

Making use of this information, we first eliminated the nine bottom-ranked methods, from *Global 5x5 kFill Filter* on down. Dropping the two kFill filters, even though they were the best of these nine, was particularly enticing because of the algorithm's running time, which rivals that of the OCR process itself. Out of the remaining five methods, *Fill Holes and Breaks; Cut* and *Global Fill Holes and Breaks* addressed the same type of document degradation. Of these two, we chose to retain the Global Fill Holes and Breaks with an eye to future extension of our methodology to variable-width fonts. The third column of Table 2 shows the number of documents for which each of this reduced set of methods caused the best OCR improvement.

What is the impact of paring the number of restoration methods from fourteen to four? When the best method out of the original fourteen is applied to each document, the OCR character error rate of the corpus drops from 20.27% to 11.34%. This decrease represents a limit on the performance of an automatic restoration method selection algorithm. When we used only the four methods in the reduced list, the OCR character error rate increases from 11.34% to 11.84%, a negligible change. Moreover, each of the four methods is best for roughly a quarter of the documents, as shown above in Table 2. This even distribution between methods will increase the accuracy of the linear classifier we will use to automatically select a restoration method for new documents.

For reasons that will become clear just below, we repeated this entire procedure using OCR *word* error rate improvement instead of *character* error rate improvement as the metric of a restoration method's success. This resulted in the same four methods' being selected, with the limit on OCR word error rate rising from 22.61% to 23.22% when we trimmed from fourteen methods down to four.

5. Automatic Restoration Method Selection

We are now in a position to train the linear classifier that will predict the best restoration method for new documents. On the one hand, we know each document's five quality measures that will be input to the classifier. On the other hand, we know the best restoration method (out of the four-method set) that will optimally improve it. More generically, we have 139 objects, each described by five features and belonging to one of four classes - a classic pattern classification problem. We therefore trained a linear

classifier, using the Pocket algorithm [13, 14], to assign each document, based on its five quality measures, to one of the four restoration methods. We did this two times, training first to the best category for improving the OCR *character* error rate, then to the best category for improving the *word* error rate.

Because of our rather small document corpus, we tested the linear classifier using cross-validation[15]. That is, we cycled through the entire corpus, on each iteration training on 138 documents and testing on the 139th. The decrease in OCR error rate resulting from the best possible restoration method for each document gave a lower bound for these results. For an upper bound, we found the outcome of choosing a restoration method randomly.

The following subsections present the results of the cross-validation test in four ways: according to improvement in OCR character accuracy, improvement in OCR word accuracy, improvement in the quality measures, and selection of the optimal OCR algorithm. By all measures, the method was a success.

5.1 OCR Character Error Rate Results

As shown in Table 3, automatic selection of a restoration method reduced the OCR character error rate of the 139-member corpus from 20.27% to 12.60%. This decreased 12.60% error rate was not quite as good as our established lower bound of 11.34% (the outcome using the best restoration method for each document), but better than our upper bound of 17.96% (from random selection of a restoration method). On our lower-quality 41-document subcorpus, the error rate fell from 34.28% to 18.82%, an equally satisfying outcome.

5.2 OCR Word Error Rate Results

As shown in Table 4, automatic selection of a restoration method reduced the OCR word error rate of the 139-member corpus from 32.17% to 24.42%. On the lower-quality 41-document subcorpus, the word error rate fell from 50.99% to 36.20%. Both results represent noteworthy improvements.

5.3 Quality Measures of Restored Documents

Even though the primary reason for restoring documents is to improve OCR accuracy, it is nonetheless interesting to ascertain if the quality measures that were used to select the restoration algorithm also show an improvement. They do indeed. Table 5 shows a marked improvement in four of the quality measures as computed across the 139-member corpus. (The fifth, font size factor, is not expected to change.) In the restored corpus, the small speckle factor improved by 44%, the white speckle factor by 40%, the touching character factor by 72%, and the broken character factor by 14%.

5.4 Selection of Optimal Algorithm

Of the 139 documents, 92 were assigned to the correct (best) restoration method after training to OCR character accuracy, and 89 were assigned to the correct restoration method after training to OCR word accuracy. In most cases, a suboptimal method still produced a meaningful enhancement to the document image in terms of decreased OCR error rate. When training to character accuracy, only fifteen of the 47 misassigned documents showed an increase in OCR error rate after restoration. Moreover, in most cases the increases were small: less than one percentage point for eight documents, between one and five percentage points for five documents, and greater than five percentage points for the final two documents. The largest increase in character error rate was from 4.66% to 14.45%, resulting from the linear classifier's erroneously choosing *Global Despeckle* instead of *Global Fill Holes and Breaks*.

Likewise, when training to word accuracy, only twelve of the 50 misassigned documents suffered an increase in OCR error rate. The magnitude of these increases was similar to those encountered in character error rates.

6. Restoration Cascade

It is tempting to couple our restoration methods in pairs. Perhaps a best restoration method would consist of background despeckle followed by a cut on the typewriter grid. We have experimented with some of these combinations and obtained spotty results. Some restoration cascades improved four or five

of our documents by an additional 10% or so character accuracy. But in general, we saw little decrease in the corpus OCR error rate as a whole.

One reason for this lackluster result may be that some of our restoration methods tend to be dual purpose already. For example, the Loce/Dougherty despeckle algorithm also tends to thin fattened strokes, as shown in Figure 8. The algorithm by itself has the effect of a cascade. Another reason the cascade

does not work is that some restoration methods also introduce artifacts into the document image. Perhaps the application of two methods in cascade introduces too many artifacts for the OCR engine to handle, and the benefit of the restoration methods is lost.

We believe that a cascade of restoration methods may still have merit; it is just too difficult to show it and train a classifier accordingly on our 139-member corpus. We will investigate the approach further when a larger 1000-member document corpus is made available to us by our funding partners.

7. Conclusions

We have presented a successful method for automatically improving the quality of document images in a typewritten archive, and we demonstrated a marked decrease in OCR error rates. Character error rates drop by about 38%, word error rates by about 24%. The method is easy to use - we view it as a pre-OCR cleanup operation, and it takes about one-tenth the computational effort of the OCR process itself. We like the linear classifier because it takes into account all five quality measures when selecting an appropriate restoration method, rather than using one or two thresholds set by trial and error on a subset of the measures.

What would be required to apply the method to a new corpus? A linear classifier trained on one corpus could be used “as is” on a new one. But if the corpora differ substantially (i.e., have different distributions of quality problems), then for optimal results, we expect that some retraining will be necessary. This would require obtaining ground truth for a training set of documents from the new corpus and rerunning the Pocket algorithm to train a new linear classifier. In this retraining process, one would not be limited to the fourteen restoration methods described in Section 4.1. Any other restoration methods

[16-18] can be included, even if they are folded into the OCR process itself, as long as they are "best" for a meaningful number of documents in the new corpus.

The effort involved in retraining the classifier, while non-trivial due to the need for ground truth, is small compared with manual selection of the best restoration algorithm for each document in a large archive. Perhaps one-time training on a very large corpus would obviate the need for repeated training on smaller specialized corpora.

We expect our methods are extensible to variable-width fonts, and we are presently pursuing that aspect of our work.

Acknowledgments

Professor Nathan Brener and his staff at Louisiana State University scanned the 139 documents used in this study and provided the keyed-text ground truth. Don Hush at Los Alamos provided assistance with the linear classifier and the Pocket algorithm. Tom Curtis, Department of Energy, and Jim Campbell and Gary Craig, Federal Intelligent Document Understanding Laboratory, provided important administrative and technical support. Technical conversations with Becker Drane and Steve Dennis, Department of Defense, were particularly valuable to us. We are indebted to a reviewer who suggested the comparison shown in Table 5.

References

1. Victor T. Tom and Paul W. Baim, "Enhancement for Imaged Document Processing," Proceedings 1995 Symposium on Document Image Understanding Technology, Annapolis, MD, p154.
2. P. Stubberud, et. al, "Adaptive Image Restoration of Text Images that Contain Touching or Broken Characters," Proceedings ICDAR'95 Third International Conference on Document Analysis and Recognition, Montreal, 1995, p778.
3. TMSSequoia, "ScanFix Software," 206 West 6th Avenue, Stillwater, OK, 74074 ©1997.

4. Venu Govindaraju and Sargur N. Srihari, “Assessment of Image Quality to Predict Readability of documents,” Proceedings, Document Image Recognition III, SPIE ’96, San Jose, CA, SPIE Vol. 2660, 1996, p333.
5. Lougheed et. al, “Method for Repairing Optical Character Recognition Performing Different Repair Operations Based on Measured Image Characteristics,” United States Patent 5,142,589, 1992.
6. Michael Cannon et. al, “An Automated System for Numerically Rating Document Image Quality,” Proceedings 1997 Symposium on Document Image Understanding Technology, Annapolis, MD, p162.
7. Luis R. Blando, et. al, “Prediction of OCR Accuracy Using Simple Image Features,” Proceedings, ICDAR’95 Third International Conference on Document Analysis and Recognition, Montreal, 1995, p319.
8. Yi Lu, “On the Segmentation of Touching Characters,” Proceedings, ICDAR’93 Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, p440.
9. Robert P. Loce and Edward R. Dougherty, Enhancement and Restoration of Digital Documents, SPIE Optical Engineering Press, 1997, p192.
10. ibid., p198.
11. Edward R. Dougherty, An Introduction to Morphological Image Processing, SPIE Optical Engineering Press, 1992, p21.
12. Lawrence O’Gorman and Rangachar Kasturi, Document Image Analysis, IEEE Computer Society Press, 1998, p13.
13. S. I. Gallant, “Preceptron-based Learning Algorithms,” IEEE Trans. Neural Networks, Vol. 1, No. 2, 1990, p179.
14. S. I. Gallant, Neural Network Learning and Expert Systems, MIT Press, 1993.
15. M. Stone, “Cross-validatory choice and assessment of statistical predictions”, Journal of the Royal Statistical Society Series B , V. 36, 1974, p111.
16. Jisheng Liang et. al, “Document Image Restoration Using Binary Morphological Filters,” Proceedings, Document Image Recognition III, SPIE ’96, San Jose, CA, January 1996, p274.

17. M. Y. Jaisimha et. al, “Model Based Restoration of Document Images for OCR,” Proceedings, Document Image Recognition III, SPIE ’96, San Jose, CA, January 1996, p297.
18. Gary E. Kopec and Mauricio Lomelin, “Document-Specific Character Template Estimation,” Proceedings, Document Image Recognition III, SPIE ’96, San Jose, CA, January 1996, p14.

FIGURES and TABLES

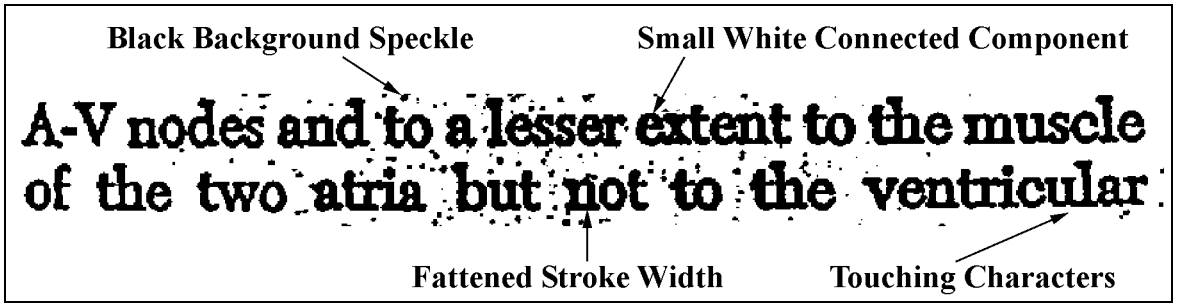

Figure 1. A portion of the eighth-generation document from our small photocopied document corpus. This corpus allowed us to study the connected-component reaction to increasing image degradation. The sample is displayed at 200% magnification.

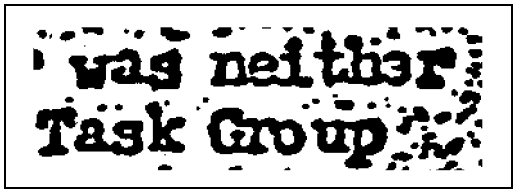

Figure 2. A portion of a page from the 139-member DOE corpus. This is a photocopy of a low-contrast carbon copy, which originally had neither background speckle nor broken characters. The photocopier automatically set a threshold to map subtle changes in gray tone to black or white.

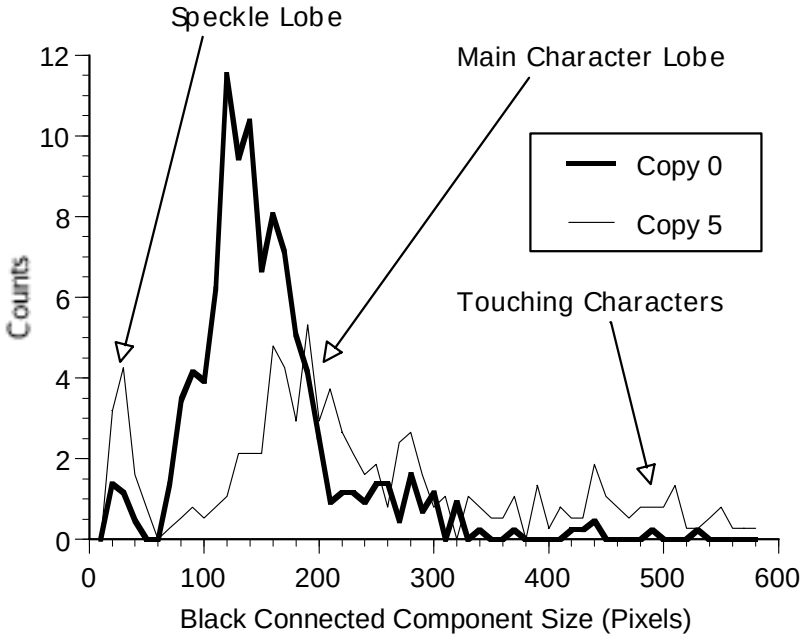

Figure 3. Two histograms of black connected component sizes computed from the zeroth and fifth generations of a photocopied document.

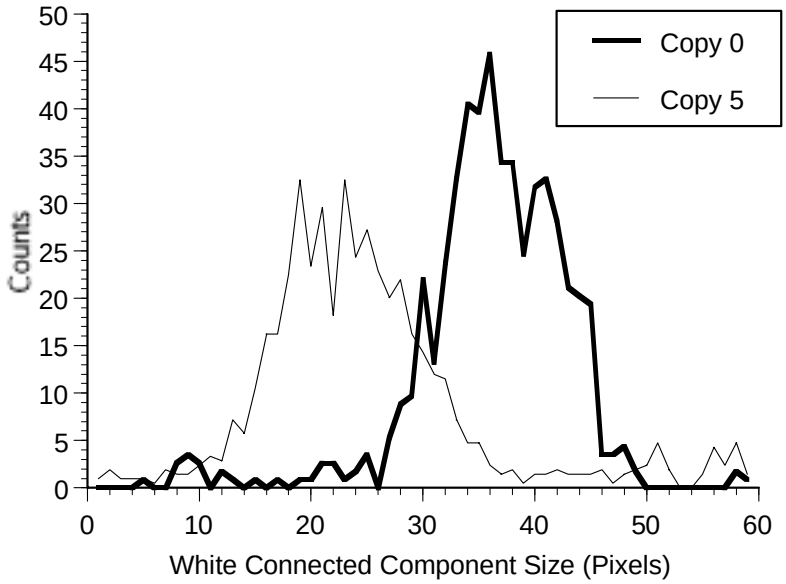

Figure 4. Two histograms of white connected component sizes computed from the zeroth and fifth generations of a photocopied document.

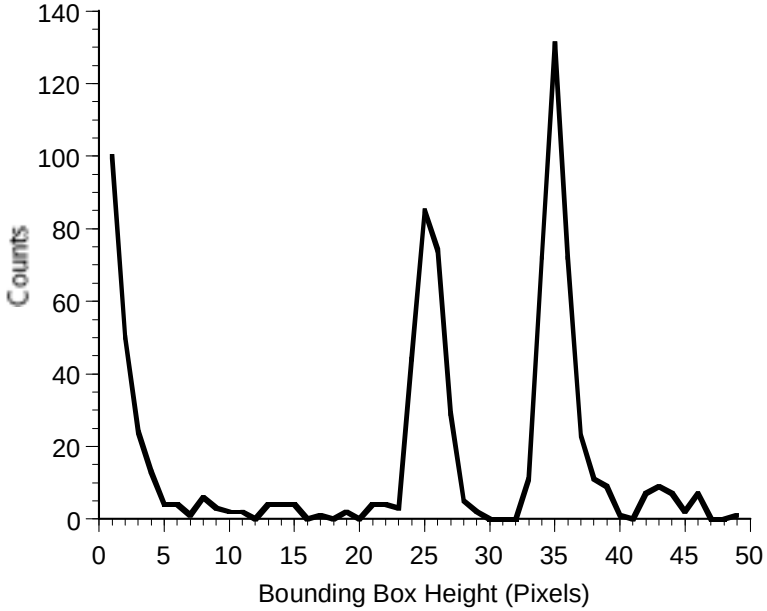

Figure 5. A typical histogram of black connected component bounding box heights. The first peak represents background speckle and is ignored. The peak at 25 pixels represents the height of typical characters having no ascenders or descenders. The peak at 36 presents characters having either an ascender or descender.

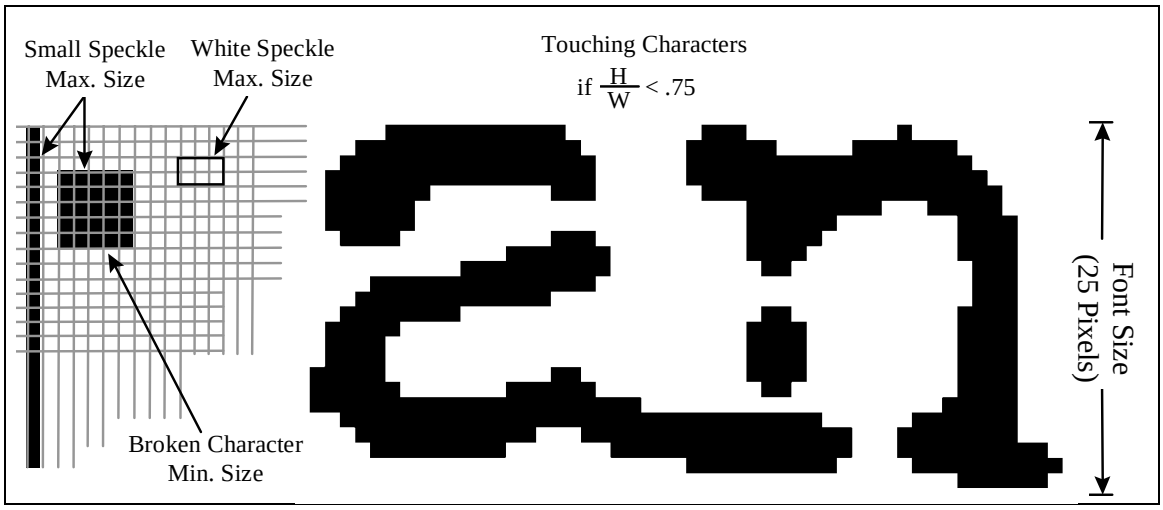

Figure 6. The intuitive nature of the quality measure parameters is shown. In this example, the font size of the characters is 25 pixels. In the computation of the *small speckle factor*, the largest acceptable black connected component is one having font size (25) pixels, and is shown as a tall, narrow, rectangle on the far left, somewhat thinner than the letter "i". Shorter, fatter components also qualify. The largest white connected component used in the computation of the *white speckle factor* is 1% of the font size squared, or 6 pixels in this example. The *broken character factor* is based on small black connected components that lie on lines of text. The smallest allowable fragment has "font size" pixels in it and must be shorter or narrower than 75% of the font size. Note that the upper fragment of the "a" qualifies, the small fragment in the "n" unfortunately does not. Last, a black connected component having an aspect ratio less than .75 contributes to the *touching character factor*. Note that in this (not uncommon) example, even though both characters are broken, a valid contribution will be made to the touching character factor.

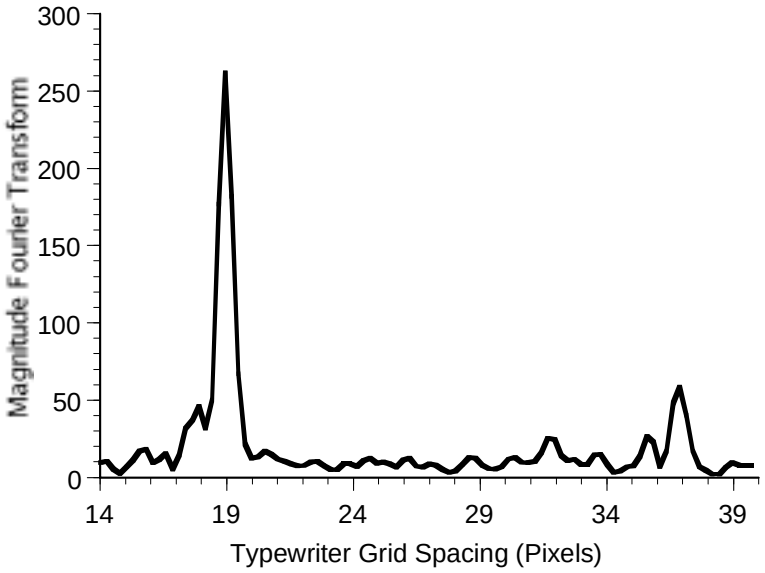

Figure 7. An average of the Fourier transforms of several lines of text. The peak at 19 pixels indicates the width of the typewriter grid.

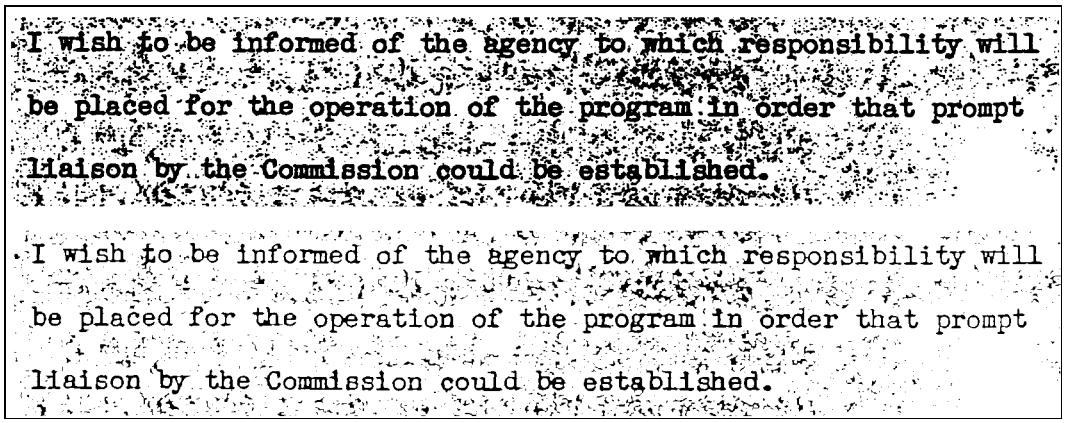

Figure 8. Top: a portion of an original document plagued by background speckle and fattened stroke widths. Bottom: the same portion of the document after enhancement by the Loce/Dougherty 2-erosion basis set. Note that both degradations have been addressed by the one enhancement method.

	SSF	WSF	TCF	BCF	FSF	OCR Error
SSF	**1.00**	**.243**	**.278**	**.176**	**-.281**	**.446**
WSF		**1.00**	**.649**	**-.375**	**-.303**	**.681**
TCF			**1.00**	**-.344**	**-.197**	**.649**
BCF				**1.00**	**.089**	**.173**
FSF					**1.00**	**-.278**
OCR Error						**1.00**

Table 1. Cross-correlations among the quality measures, and between them and OCR character error rate.

Name of Restoration Method	Number of Times Best (all 14 methods used)	Number of Times Best (4 methods used)
Global Aggressive Despeckle	**29**	**32**
Global Despeckle	**24**	**35**
Fill Holes and Breaks; Cut	**21**	
Cut on Typewriter Grid	**19**	**28**
Global Fill Holes and Breaks	**15**	**44**
Global 5x5 kFill Filter	**10**	
Global 4x4 kFill Filter	**8**	
Despeckle; Cut	**4**	
Aggressive Despeckle; Cut	**4**	
Do Nothing	**3**	
Global 3x3 kFill Filter	**2**	
Global 3x3 Morphological Close	**0**	
Global 5x5 Morphological Close	**0**	
Global 7x7 Morphological Close	**0**	
Total Documents	**139**	**139**

Table 2. The number of times each of our restoration methods was best on the 139 documents in our typewritten corpus. Results are shown separately for the original list of fourteen methods (column 2) and the pared list of four methods (column 3)

Corpus Description	139 Documents	41 Documents
No restoration	**20.27%**	**34.28%**
Random selection of four restoration methods	**17.96%**	**27.48%**
Linear classifier selection of four methods	**12.60%**	**18.82%**
Best of four restoration methods	**11.84%**	**17.49%**
Best of fourteen restoration methods	**11.34%**	**16.84%**

Table 3. A compilation of OCR character error rates resulting from a variety of restoration method selection criteria. Results are shown for our entire 139-member typewritten corpus, as well as a lower-quality 41-member subcorpus.

Corpus Description	139 Documents	41 Documents
No restoration	**32.17%**	**50.99%**
Linear classifier selection of four methods	**24.42%**	**36.20%**
Best of four restoration methods	**23.22%**	**34.38%**
Best of fourteen restoration methods	**22.61%**	**33.77%**

Table 4. A compilation of OCR word error rates resulting from a variety of restoration method selection criteria. Results are shown for our entire 139-member typewritten corpus, as well as a lower-quality 41-member subcorpus.

Quality Measure	Original	Restored
Small speckle factor	**.192**	**.107**
White speckle factor	**.114**	**.068**
Touching character factor	**.242**	**.067**
Broken character factor	**.143**	**.124**

Table 5. A comparison of quality measure values before and after document image restoration. The values shown are average values computed across our corpus of 139 documents. Individual factors were weighted by the number of characters in a document; the average was normalized by the total number of characters in the corpus. The decrease in the magnitude of the individual quality measures after restoration indicates a quantifiable improvement in document quality.